

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
28 February 2002 (28.02.2002)

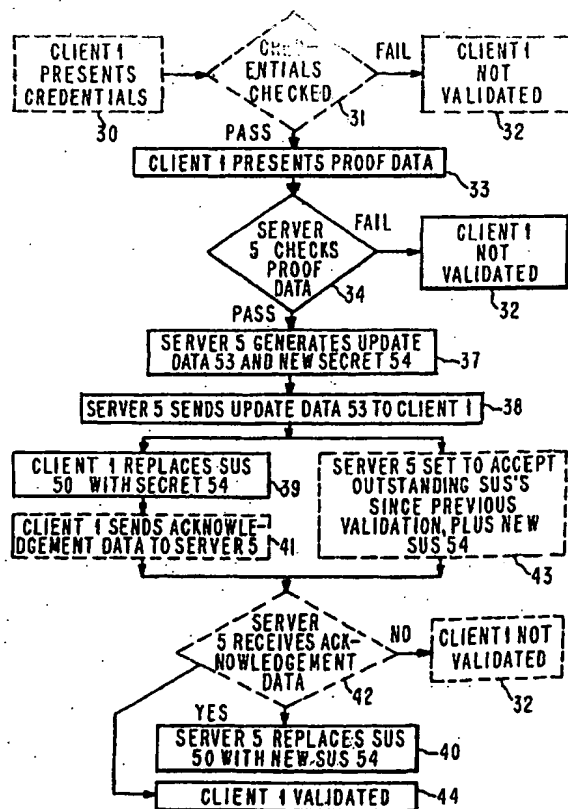
PCT

(10) International Publication Number
WO 02/17555 A2

- (51) International Patent Classification⁷: H04L 9/08
- (21) International Application Number: PCT/US01/25927
- (22) International Filing Date: 16 August 2001 (16.08.2001)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
60/226,429 18 August 2000 (18.08.2000) US
09/921,265 1 August 2001 (01.08.2001) US
- (71) Applicant: VERISIGN, INC. [US/US]; 1350 Charleston Road, Mountain View, CA 94043 (US).
- (72) Inventor: FORD, Warwick; 6 Ellery Square, Cambridge, MA 02138 (US).
- (74) Agents: RADLO, Edward, J. et al.; Fenwick & West LLP, Two Palo Alto Square, Palo Alto, CA 94306 (US).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: COUNTERING CREDENTIALS COPYING



(57) Abstract: Systems, methods and computer readable media for enabling a server device (5) to validate one or more client devices (1). A shared unpredictable secret (50) is generated. The shared unpredictable secret (50) is stored in the client device (1) and in the server device (5). The client device (1) proves possession of a correct shared unpredictable secret (50) to the server device (5). The shared unpredictable secret (50) is replaced by a new shared unpredictable secret (54) each time the client device (1) is validated by the server device (5).

WO 02/17555 A2

These and other more detailed and specific objects and features of the present invention are more fully disclosed in the following specification, reference being had to the accompanying drawings, in which:

5 Figure 1 is a block system-level diagram of a preferred embodiment of the present invention.

Figure 2 is a flow diagram of a preferred embodiment of the registration/reset phase of the present invention.

10 Figure 3 is a flow diagram of a preferred embodiment of the log-in phase of the present invention.

Figure 4 is a flow diagram illustrating an alternative embodiment of the method illustrated in Figure 3.

Figure 5 is an illustration of shared unpredictable secret 50 and its progeny.

15 Detailed Description of the Preferred Embodiments

As used in the present patent application, client (sometimes referred to as "client device") 1 can be any digital device, e.g., a personal computer, mobile phone, smartboard, Internet appliance, or any other network accessible device.

20 There may be one client 1 or, as illustrated in Figure 1, a finite number n of clients 1. Each client 1 wishes to communicate with an infrastructural component that is referred to in the present patent application as a server (or "server device") 5. Server 5 may provide any type of service to client 1. For example, server 5 might be an Internet service provider or a telephone network access point. The communications link between client 1 and server 5 may be any link, such as a wireless link, a wired link, or a link over a network 4, which may be an open network such as the Internet. The process that
30 commences with client 1 requesting validation with server 5 and including server 5 checking proof data that client 1 has generated from the shared unpredictable secret 50 is referred to herein as a "log-in". A log-in results in client 1 being either validated or not validated by server 5.

One concern in such an environment pertains to credentials sharing. In this scenario, a person who has access to a client device 1 voluntarily shares his personal credentials, such as a password or private cryptographic key, with other user devices 2. All of these user devices 2 employ the user account of the original user. Two problems that arise from this scenario are: 1) It is difficult for server 5 to hold particular users 2 accountable for their actions when using the services provided by server 5, since some or all users 2 are indistinguishable from each other; and 2) Users may fraudulently avoid paying subscription fees that are designed for payment on a per-user basis.

Another concern is outright credentials theft. In this scenario, a nefarious person having access to a client-like device, referred to as "attacker 3" in Figure 1, penetrates a legitimate client device 1, copies stored credentials data from client device 1 into attacker device 3, perhaps supplements this thievery with a determination of other information such as the user's password, personal identification number, or social security number from other sources, and then masquerades as the legitimate user from the attacker device 3. When the client device 1 being attacked is a hardware device and not a software module, this scenario is sometimes referred to as "device cloning". Client devices 1 that are typically cloned include mobile telephones and smartcards.

The present invention also applies to the copying of a portable credentials storage medium such as a magnetic stripe card or ticket if that medium is writable as well as readable by the terminal device in which it is used. In this case, the term client device 1 is taken to mean the combination of the portable credentials storage medium and the terminal device in which it is currently inserted.

The present invention uses a method of stateful authenticators to provide a low cost, low overhead means of detecting when one user account is being employed for more than

At step 22, server 5 decides whether the authentication data presented by client 1 are acceptable. If not, client 1 is not allowed to register (step 26). If acceptable, client 1 is allowed to register (step 23). In this eventuality, the shared unpredictable secret 50 is generated (step 24).

As illustrated in Figure 5, the shared unpredictable secret (SUS) 50 comprises an unpredictable component 51 and an optional fixed component 52. Unpredictable component 51 may be generated by a random number generator or a pseudo-random number generator. Typically, unpredictable component 51 is generated at server 5 and communicated to client 1, but it may also be generated at client 1, or at a combination of server 5 and client 1. As an alternative to communicating unpredictable component 51 to client 1, unpredictable component 51 may be pre-installed in client device 1 during manufacture or during a device 1 personalization process.

When used, fixed component 52 typically comprises identification information. For example, fixed component 52 may be a serial number of the client device 1. This could be useful when there are two or more client devices 1 associated with the same user account number. Conversely, fixed component 52 could be the user account number when there is more than one user account number associated with the same client device 1. This situation may occur, for example, when a user has one account number for business use and another account number for personal use; or when two users share the same cellular telephone 1; or when the client device 1 is a terminal into which users insert their credentials storage media.

Typically, after server 5 has generated SUS 50, server 5 transmits SUS 50 to client 1. The transmission is preferably encrypted, for reasons of security. Any type of encryption, including symmetric or asymmetric encryption, may be used. Alternatively, an SUS 50 having an unpredictable component 51 is generated by the aforementioned Diffie-Hellman key exchange technique, or pre-installed in device 1 as described

subsequent log-in following the re-set of server 5 as described in conjunction with step 43, below. For purposes of simplifying this discussion, it will be assumed that the proof data are computed upon SUS 50.) It is desirable that SUS 50 itself be not directly communicated over an open network 4 lest it be intercepted by a nefarious person. One method by which client 1 computes proof data without revealing SUS 50 itself is for client 1 to compute a one-way function of SUS 50. The one-way function is typically a cryptographic hash function. Then, at step 34, server 5 checks the proof data by applying its (server 5's) proof data generation algorithm to its (server 5's) stored version of SUS 50. If the proof data generated by server 5 matches the proof data presented by client 1, client 1 has passed the test, and the method proceeds to step 37. Otherwise, client 1 has failed the test and is not validated at step 32. Step 32 operates as previously described.

In order for this method to work, client 1 and server 5 have to be using consistent if not identical proof data generation algorithms. It is immaterial whether or not an eavesdropper knows what this algorithm is (or what these algorithms are).

At step 37, update data 53 and a new shared unpredictable secret 54 are generated, typically by sever 5. On Fig. 3 (boxes 37 and 39), item 54 is referred to as a "secret" rather than a "shared unpredictable secret", because the contents of the storage area that server 5 uses for SUS's are not replaced with new SUS 54 until step 40 is executed, below. Thus, the secret is not yet "shared".

Then, at step 38, server 5 sends update data 53 to client 1. Update data 53 is such that client 1 and server 5 are able to generate new SUS 54 from the most recent version of SUS 50, by means of applying update data 53 thereto. Typically, the step of applying update data 53 to SUS 50 comprises computing a one-way function of the combination of SUS 50 and update data 53. For example, update data 53 could be a new random or

has received the update data 53, but for some reason the ACK has been lost in transit, during the next log-in, client 1 at step 33 will be presenting proof data emanating from new SUS 54, and server 5 at step 34 will be programmed to accept said data. If, on the other hand, update data 53 were not received by client 1, then, during the next log-in, client 1 at step 33 will be presenting proof data emanating from an old SUS 50, and server 5 at step 34 will be programmed to accept said data.

Step 42 illustrates the reality that server 5 may or may not receive the ACK, either because the update data 53 were lost in transit, the ACK was lost in transit, or the client device 1 failed. If server 5 receives the ACK within a preselected "time-out" period, then client 1 is validated at step 44 and step 40 is entered into. At step 40, server 5 updates its (server 5's) storage area that is allocated to SUS's with new secret 54. Thus, new secret 54 becomes a new shared unpredictable secret, because it is now shared by client 1 and server 5. As a substep to step 40, server 5 deletes SUS 50 and any older SUS's from its list of acceptable SUS's. Thus, in future log-in attempts, both client 1 and server 5 will have stored therewithin new SUS 54; and the proof data (of client 1 in step 33 and server 5 in step 34) will be based upon new SUS 54.

If, on the other hand, server 5 does not receive at step 42 the ACK from client 1 during the time-out period, client 1 is not validated at step 32, which executes as described previously.

The protocols described herein have the following desirable properties: 1) Any SUS (50) value produces no more than one validation; and 2) If the protocol fails at any stage, both client 1 and server 5 are left in a state that a new invocation of the protocol will operate correctly.

Optionally, for example in conjunction with step 34, server 5 maintains an audit trail of log-in attempts, noting in particular those log-in attempts in which the step 34 checks

2) The attacker 3 logs in before legitimate client 1 logs-in again. In this case, attacker 3 can successfully masquerade as a legitimate user up to the time of the legitimate user's next log-in. On the legitimate user's next log-in, server 5 will be alerted and special action taken. This special action might include out-of-band communication with the legitimate user, investigation of the situation, and consequent shut-out of attacker 3 from further validations.

An alternative embodiment is illustrated in Figure 4. The method of claim 4 is identical to the method of Figure 3 wherein acknowledgement data are used, except that steps 33 and 34 are consolidated into steps 41 and 42. Thus, in the Figure 4 embodiment, unlike the Figure 3 embodiment, the presentation of proof data by client 1 and the checking of the proof data by server 5 are not performed prior to server 5 generating update data 53 and new secret 54 in step 37.

Step 50 of Figure 4 replaces step 41 of Figure 3. In step 50, client 1 sends both the proof data and the acknowledgement data to server 5. The proof data are computed on the new secret 54. The proof data could serve a double role as proof data and acknowledgement data. In this case, there is no need for acknowledgement data separate and apart from the proof data.

Step 51 of Figure 4 replaces step 42 of Figure 3. In step 51, server 5 checks both the proof data and the acknowledgement data, or, in the case where the proof data serve the double role as proof data and acknowledgement data, the proof data.

As stated previously, there may be legitimate use of a number n of different client devices 1 by a single legitimate user. In this case, server 5 holds current SUS's 50 and new SUS's 54 for each client device 1, and considers each client device 1 to be legitimate; and each client device 1 has its own unique SUS 50 and new SUS 54. The number n of clients 1 may be restricted in accordance with local policy. In this scenario, SUS's 50, 54 are respectively unique from one device 1 to the

Claims

1. A method for validating a client device by a server device; said method comprising the steps of:
 - generating a shared unpredictable secret;
 - 5 storing the shared unpredictable secret in the client device and in the server device;
 - requiring the client device to prove that it holds a correct secret as a precondition to the server device validating the client device; and
 - 10 replacing the shared unpredictable secret by a new shared unpredictable secret when the server device validates the client device.
2. The method of claim 1 wherein an initial shared unpredictable secret is determined in the client device and in
15 the server device during a registration step that occurs prior to a log-in step.
3. The method of claim 2 wherein the registration step entails more checking of bona fides of the client device than does a log-in step.
- 20 4. The method of claim 2 wherein, during the registration step, the client device is required to make a payment to the user device.
5. The method of claim 1 wherein the shared unpredictable secret is generated by a generator from the group comprising a
25 random number generator and a pseudo-random number generator.
6. The method of claim 1 wherein the shared unpredictable secret comprises an unpredictable component and a fixed component.
7. The method of claim 1 wherein a plurality of client
30 devices desire to be validated by the server device; and each client device has a unique unpredictable secret that it shares with the server device.
8. The method of claim 1 wherein, following a validation of the client device, the server device discards the original
35 shared unpredictable secret and stores within the server device

the server device is adapted to accept from the client device any proof data that are generated from a secret that is newer than the secret for which the most recent acknowledgment data have been received by the server device.

14. The method of claim 11 wherein:

the client device sends to the server device both the acknowledgment data and proof data derived from the new secret.

15. The method of claim 14 wherein:

the proof data are computed on the new secret; and the proof data serve also as acknowledgment data.

16. The method of claim 1 wherein:

the client device presents proof data to the server device, wherein the proof data are derived from a shared unpredictable secret using a proof data generation algorithm, and the proof data do not divulge the shared unpredictable secret;

the server device checks the proof data by using a proof data generation algorithm consistent with the proof data generation algorithm used by the client device; and

when the server device determines that the proof data presented by the client device were not generated from the same shared unpredictable secret that is stored in both the client device and in the server device, the server device does not validate the client device.

17. The method of claim 16 wherein each proof data generation algorithm is a one-way function.

18. A system for enabling a server device to validate a client device, said system comprising:

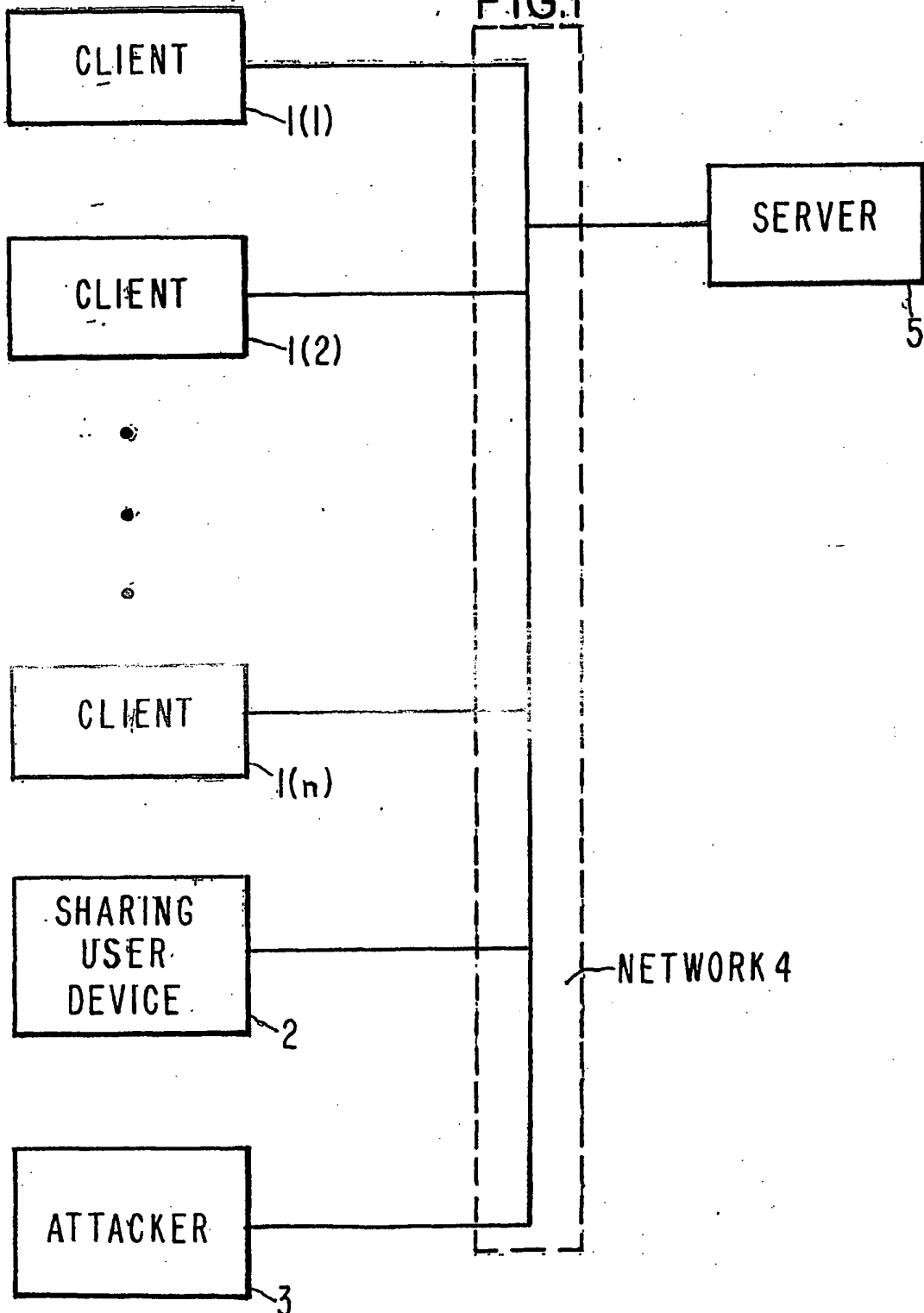
at least one client device;

a server device;

a shared unpredictable secret;

1/5

FIG. 1



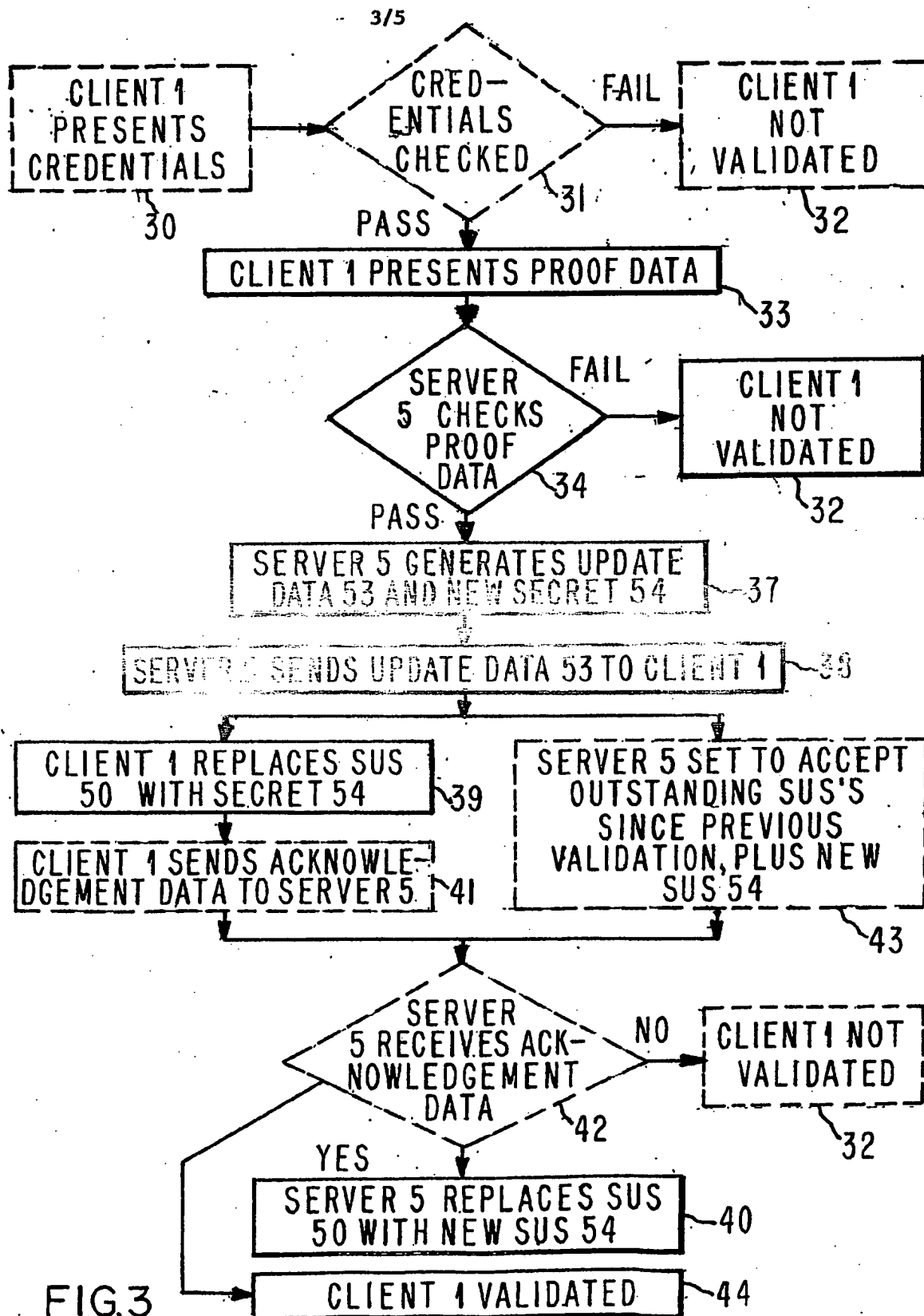


FIG.3

5/5

FIG. 5

